THE SEED

January 1986
Vol. 8 No. 1

NEWSLETTER FOR APPLE PEOPLE

APPLE PI DENVER

PRESIDENT'S REPORT

Well, we are into a new year. I hope this one turns out to be a great one for everyone. In the past 12 months, Apple Pi has grown and become the standard in Denver for people who seek out help with their Apple computers. In 1985 we started a second Core Meet for the people who live or work in the south end of town, and people are asking for another to be started in the northeast. We have seen more schools take advantage of our school associate member program and have had tremendous support from the Thornton High School Computer Club.

At this time I would like to thank the officers for the great job they have done for the first half of their terms. Ed Thompson, our Librarian, has put in hundreds of hours organizing the libraries as well as adding lots of great programs. The DOMs have been super --- don't miss out on getting yours each month. Roy Dean, our Store Keeper, has done a wonderful job of getting great goodies at terrific prices. Joann Bachman, our Treasurer, has kept us out of the red, and Mary Ann Vitry, our Secretary, is doing a good job of keeping track of you all. Tom Angelos, your Vice President, has been doing a dandy job of finding guest speakers for the meetings, and Trent Hein answers more of your questions than just about everyone. Steven Martin has kept the BBS chugging along nicely, and Lianne Nusse and Bob Hein keep up rolling with all the tasks that they perform. Next time you say hi to one of these folks, thank them for all the time they are putting in, and if you can help in any of the club tasks, PLEASE LET US KNOW! To all the officers and other helpers I have not mentioned, and to all at Apple Pi, HAVE A GREAT 1986!!!

Randy

SHAPE TABLES AND ANIMATION..... 3
LIBRARIAN'S REPORT..... 8
JOHNNY APPLESEED..... 9
PRODOS TECHNICAL NOTES 13 AND 14..... 10

denver apple pi Seed

P.O. BOX 17467
DENVER, COLORADO 80217

President . . . . . .Randall Ansley
Vice President . . . . . .Tom Angelos
Secretary . . . . . .Mary Ann Vitry
Treasurer . . . . . Joann Bachman
IAC Representative & Member of the Board . . . . . .Trent Hein
Sales Officer & Member of the Board . . . . . .Roy Dean
Software Librarian & Member of the Board . . . . . Ed Thompson
Store Representative & Facilities / Property Officer . . Bob Hein
New Member Contact Officer . . . . . .Lianne Nusse
BBS Sysop . . . . . .Steven Martin

MEMBERSHIP INFORMATION

APPLE PI is an Apple Computer Users Group - not affiliated with Apple Computer, Inc. - that provides many member benefits, some of which are:

MONTHLY MEETINGS - On the first Thursday of each month at 7:00 p.m. APPLE PI holds its general meeting at the Green Center, Colorado School of Mines Campus, Golden, Colorado. This meeting is a combination of an informational meeting, new product announcements, and a short business meeting. The members are given an opportunity to buy computer supplies a little above cost. Software products are given away as door prizes.

On the Sunday following the general meeting there is an informal meet which is usually held at the Clear Creek Office Park, 4251 Kipling, in Wheat Ridge. This meeting is from 10:00 a.m. until 3:00 p.m. and provides a more informal environment to seek or give advice to fellow members and get copies of programs from the APPLE PI Library. Both of these meetings are open to the general public but only members have access to the club library and products that are sold.

PROGRAM LIBRARY - APPLE PI currently has a library of 9,000 programs. All current disks are in 16 sector (3.3, Pascal, CP/M) format and are available to any member. APPLE PI Members can obtain copies from the library:

1) By coming to the monthly informal meeting (swap meet);
2) By written request to the club librarian (at any meeting or by mail);
3) By electronic mail on APPLE PI's bulletin board system:

DISK OF THE MONTH - Each month a collection of some of the best programs donated to APPLE PI during the preceding month are made available. The DOM can be purchased at any group meeting or by mail.

DUES - Current membership dues are **$18.00** per year, with a one-time application fee of **$10.00.** Checks should be made payable to DENVER APPLE PI.

The words APPLE, APPLE COMPUTER, and the APPLE LOGO are registered trademarks of the Apple Computer Company Inc. Representation of these trademarks in the Seed does not constitute any expression of agreement or disagreement by Apple Computer Inc. with any articles or other printed material found in the Seed. The opinions, policies and comments contained in the Seed are those of the Apple Pi users group, officers and members.

NOTICE

Articles, stories, and other items in the Seed may be copied or used by other clubs or groups as long as credit is given to Denver Apple Pi and the author of the article. On reprints please include the date or volume number from the Seed.

ADVERTISING RATES

1/4 Page	$ 25.00
1/2 Page	$ 40.00
Full Page	$ 70.00

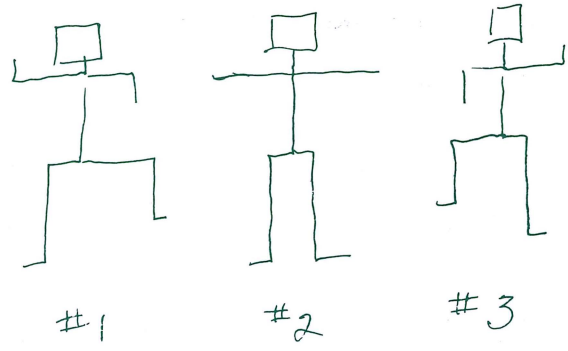
SHAPE TABLES AND ANIMATION

by Peg Smith

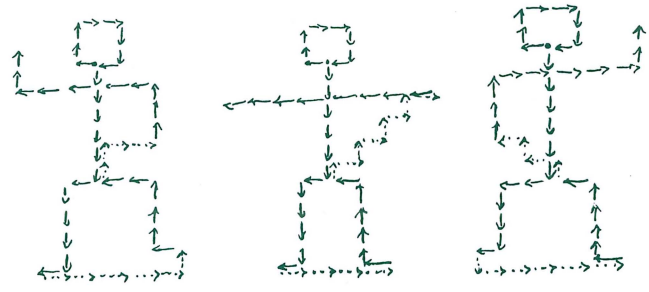
You've had your computer at least three days, and in that time you've learned quite a bit about programming in Applesoft. You've even found time to learn high and low resolution graphics, and "drawn" a few pictures. You're ready to write the Great American Arcade Game, but the thought of all the HPLOTs it will take to do the graphics is intimidating, to say the least. You've kept your ears open at Apple Pi meetings, though, and heard about something called a shape table, which is supposed to have something to do with graphics - rumor has it that a shape table will allow you to draw shapes more quickly than you could with a series of HPLOT statements which must be interpreted before each execution. You know from your earlier experience with hi-res graphics that a picture drawn with HPLOTs doesn't appear on the screen all at once, and you want the figures in your game to appear instantaneously (or at least so quickly that the user doesn't have to sit and wait while the figure is outlined). Now that you think about it, you also want your figures to be able to move. Although this article does not pretend to teach Everrything You Always Wanted to Know About Shape Tables and Animation, But Were Afraid to Ask, it will give you a basic idea of how to create your own shape tables, and how to use them to create animated graphics. Of course, there are programs (shape table makers) that do a lot of this for you, but I just KNOW that you want to be able to Do It By Yourself! We'll leave the big question of how a shape table actually WORKS for another time, and just learn how to BUILD a shape table today.

The idea behind animation isn't very difficult, and you've probably already had some experience in this area, whether you remember it or not.
Think back to the time in grade school when you just couldn't pay attention in class any more, so you drew a series of cartoons on the lower right hand corners of the pages of your textbook. You could flip the pages quickly with your thumbnail as you sat in the back row, looking (you hoped) studious, and entertain yourself with your private animated cartoon. To do animation on your Apple, you will do essentially the same thing - draw a series of shapes, just as you did in grade school, and display them on the screen, one after another. There's only one difference. In the grade school example, each page of the book covered up the preceding sketch, and when you try to simulate this on a computer screen, you'll have to erase each sketch before displaying the next one. Don't worry, there's an easy way to do this.

The first step is to draw the series of shapes you want to use. Before you start in on a great Work Of Art, you need to be aware that you must draw your shape using ONLY horizontal and vertical line segments - like stick figures without any diagonal lines or curves. The example I'll use is The Marching Man. I need three sketches to make him march - one with his left foot raised, one with both feet on the ground, and one with his right foot raised:



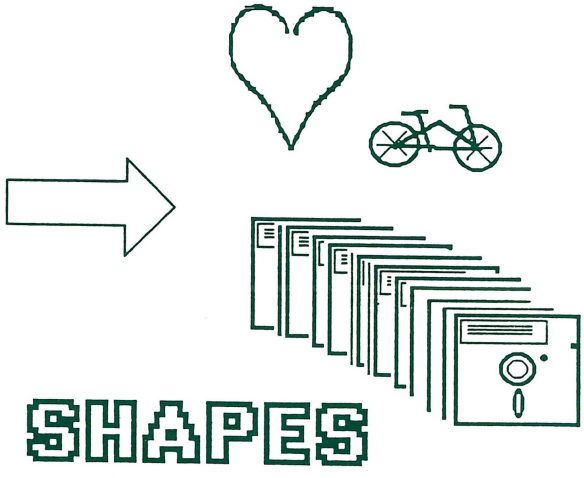
First I sketch a rough outline of the shapes I want, and then I redraw them carefully using horizontal and vertical vectors (don't panic, just a math word which tranlates LOOSELY to "arrows"; you need to know this word because some of the books you may refer to for more information on graphics will use the term) of uniform length. Once again for those of us who only speak English - we have to tell the computer two things: whether to draw at a particular position, and what direction to move from there. In the diagrams below, a solid arrow indicates plotting on the screen and moving in the direction indicated. A dotted arrow indicates moving in the direction shown WITHOUT plotting. It's kind of like those puzzles where you had to draw a figure without lifting your pencil from the paper, except that here we can avoid retracing a line by using "move without plotting" vectors (there's that word again!). Be SURE to use the same starting point for each of your figures, or the animation will not work out as easily (or at all).



The artistic part is done, and now we're ready to translate all those little arrows into a real, live shape table. Each type of arrow (vector) has a specific code, and we must write down the codes in the order in which the vectors were drawn.

UP WITHOUT PLOTTING	000
RIGHT WITHOUT PLOTTING	001
DOWN WITHOUT PLOTTING	010
LEFT WITHOUT PLOTTING	011
UP WITH PLOTTING	100
RIGHT WITH PLOTTING	101
DOWN WITH PLOTTING	110
LEFT WITH PLOTTING	111

The easiest way to write the codes in the right order is to list the vectors in a long column, starting with the first one you drew and working your way around the figure in the order you originally drew it. Then go back and write the codes next to the vectors.



SHAPE #1	SHAPE #2	SHAPE #3
111	111	111
100	100	100
100	100	100
101	101	101
101	101	101
110	110	110
110	110	110
111	111	111
110	110	110
110	110	110
110	110	110
110	110	110
110	110	110
110	110	110
110	110	110
111	111	111
110	110	111
110	110	110
110	110	110
110	110	110
111	111	111
001	001	010
001	001	001
001	001	001
001	001	001
001	001	001
001	111	001
000	100	001
111	100	111
100	100	100
100	100	100
100	111	100
111	000	100
111	001	111
000	000	000
000	001	011
001	000	000
001	001	011
100	000	100
100	001	100
100	111	100
111	111	101
111	111	101
111	111	101
111	111	101
111	111	101
100	111	100
100	111	100

Are we finished yet? Not quite. Now we have to organize these codes into BYTES (8-digit binary numbers) which the computer will read to determine which pixels to turn on and which to turn off. At this point, you don't need to UNDERSTAND all the ins and outs of this procedure, but you do need to LEARN the PROCESS. We start by making a chart with three columns labelled (from left to right) C, B and A. Columns A and B may contain three digits. Column C may contain only two digits, because one BYTE has only eight digits (3+3+2=8). Go down your list of codes, placing one code into each column in the order A,B,C,A,B,C, etc, as long as the code fits into the column. THIS IS IMPORTANT!! When you are deciding if a code can fit into column C (the one with only two digits), it's OK to put a code that starts with 0 into column C. The first zero will be lost, but the computer ignores LEADING zeroes anyway, so we have not lost anything (010 and 10 represent the same number). DON'T try to put a code such as 110 into column C, though. The first "1" will be lost, and that CHANGES the code from "move down WITH plotting" to "move down WITHOUT plotting".

There is one other problem that can arise because of the fact that the computer ignores leading zeroes. You have to be very careful where the code 000 (move up without plotting) is stored. It CANNOT be placed into column C (even though it will fit) because it will look like a series of leading zeroes and will be ignored. It can be placed in column B only if there will be a non-zero code in column C, or again it will be read as a series of leading zeroes. The same is true of placing 000 in column A. This means that you usually cannot have a series of "move up without plotting" codes one after another, because they will look like leading zeroes. The example I'm using has a few instances in which this occurs, so that you can see how to deal with the problem. Notice that I've left a space between the second and third digits of the code in column B. I'll explain the reason for this in a minute.

C	B	A	
	10 0	111	
	10 1	100	
	11 0	101	
	11 1	110	
	11 0	110	
	11 0	110	
	11 0	110	
	11 0	111	
	11 0	110	
01	11 1	110	leading zero dropped in col. C
01	00 1	001	leading zero dropped in col. C
	00 0	001	HERE's OUR FIRST DISASTER.

The zeroes in column B in the line above will not be interpreted as "move up without plotting". They will be ignored as leading zeroes. TO avoid the problem, we rewrite this code in column A of the next line, and continue as before:

C	B	A	
	10 0	111	
	10 1	100	
	11 0	101	
	11 1	110	
	11 0	110	
	11 0	110	
	11 0	110	
	11 0	111	
	11 0	110	
01	11 1	110	leading zero dropped in col. C
01	00 1	001	leading zero dropped in col. C
	00 0	001	HERE's OUR FIRST DISASTER.
	11 1	000	We recover from the disaster.
	10 0	100	
	11 1	100	
01	00 0	111	the 000 is OK in this line
	10 0	001	
	10 0	100	
	11 1	111	
	11 1	111	
	10 0	111	
		100	

Now for the reason we left the blank space between the second and third digits in column B. Draw a vertical line through these spaces. You have just divided the BYTEs you created in half, into a quantity called a NIBBLE (a nibble is half a byte - I don't make up these words, I just use them).

C	B	A	
	10 0	111	27
	10 1	100	2C
	11 0	101	35
	11 1	110	3E
	11 0	110	36
	11 0	110	36
	11 0	110	36
	11 0	111	37
	11 0	110	36
01	11 1	110	7E
01	00 1	001	49

	00	0	001	01
	11	1	000	28
	10	0	100	24
	11	1	100	3C
01	00	0	111	47
	10	0	001	21
	10	0	100	24
	11	1	111	3F
	11	1	111	3F
	10	0	111	27
			100	04
				00

Computers are strange beasties. They're never satisfied, and always seem to want YOU to do more of the work! You've spent all this time entering codes in binary format (those numbers using only 1's and 0's as digits), and now you'll have to translate into the HEXADECIMAL number system to enterr all this stuff into the computer's memory. For the time being, don't worry about what binary and hexadecimal (hex for short) really are. We'll just use a chart to make the conversion (results are shown above). The digits to the left of the line are translated to make the first digit of the hex value, and the digits to the right of the line are translated to make the right digit of the hex number. Finally, a code of 00 is added at the end to tell the computer that this is the end of the shape. If you forget to do this, you'll get really strange results, as the conputer tries to draw pictures based on whatever garbage is in its memory.

Here are the binary to hex conversions:

0000 = 0	1000 = 8
0001 = 1	1001 = 9
0010 = 2	1010 = A
0011 = 3	1011 = B
0100 = 4	1100 = C
0101 = 5	1101 = D
0110 = 6	1110 = E
0111 = 7	1111 = F

Now do the same thing for the other two shapes.

SHAPE #2:

C	B	A	
10	0	111	27
10	1	100	2C
11	0	101	35
11	1	110	3E
11	0	110	36
11	0	110	36
11	0	110	36
11	0	111	37
11	0	110	36
11	1	110	3E

01	00	1	001	49
	11	1	001	39
	10	0	100	24
	10	0	100	24
01	00	0	111	47

If we were to code the next three vectors in the order drawn, the third vector, 000, would be ignored. Since we're only moving without plotting at this point, we can just switch the second and third vectors in this set to avoid the problem.

01	00	0	000	40
01	00	0	000	41
	11	1	111	3F
	11	1	111	3F
	11	1	111	3F
	11	1	111	3F
				00

SHAPE #3:

C	B	A	
10	0	111	27
10	1	100	2C
11	0	101	35
11	1	110	3E
11	0	110	36
11	0	110	36
11	0	110	36
11	1	111	3F
11	0	110	36
11	1	110	3E
01	00	1	010	4A
01	00	1	001	49
	10	0	111	27
	10	0	100	24
	11	1	100	3C

If the next three vectors are coded in the order drawn, the third will be ignored. Again, these are draw without plotting vectors, so we can make a slight change without causing any trouble. If you look at your original vector list, you'll see that we can exchange that third vector with the one right after it (the fourth vector). After doing this, we continue as follows:

11	01	1	000	D8
	10	0	000	20
	10	0	100	24
	10	1	101	2D
	10	1	101	2D
	10	0	101	25
			100	04
				00

Now we're ready to actually create the shape table. First we must count the number of bytes (including the end-of-shape byte, 00) in each shape. Shape #1 has 23 bytes, Shape #2 has 22 bytes, and Shape #3 has 23 bytes.

The SHAPE DIRECTORY contains these things:
number of shapes
offset of shape one from beginning of directory
(number of bytes in directory)
this will be 8 for this shape table- trust me
offset of shape two from beginning of directory
(#bytes in directory + number bytes in shape #1)
8 + 23 = 31
offset of shape three from beginning of directory (#bytes in directory + # bytes in shape #1 + # bytes in shape 2)
8 + 23 + 22 = 53

These numbers must be changed to hexadecimal values - divide by 16. The quotient is the first digit and the remainder is the second digit. In hex, the digit for 10 is A, 11 is B, 12 is C, 13 is D, 14 is E, and 15 is F, so:

8=8
31=1F
53=35

Now we are ready to store the shape table in the computer's memory (yes, you have to turn the Apple on). From Applesoft, type CALL-151 and press RETURN. You'll see an asterisk prompt. You are now "in the Apple Monitor", talking to the computer in its own language. You will type 300: followed by the values in the direcctory and the codes for your shape table. The directory values are:

03 number of shapes
00 separator
08 offset for shape 1
00 would have a non-zero value only if the offset were more than 255
1F offset for shape 2
00
35 offset for shape 3
00
now type the shape codes

Your entry will look like this:

300:03 00 08 00 1F 00 35 00 27 2C 35 3E 35 etc.

Notice that you must have exactly one space between values. Shape #1 began with 27, and you will type in all the codes for shape #1, ending with 00, then all the codes for shape #2, ending with 00, and finally all the codes for shape #3. Then press return. To check your typing, type 300L, check this screen, type L again for the next page and so on. Make any necessary corrections. To get back to Applesoft, type CTRL-C RETURN. Now save the shape table with the command

BSAVE MARCHING MAN.PIC,A$300,L76

The A$300 tells the computer the address in memory of the shape table, and the L80 tells it how many bytes to save (8+23+22+23=76).

Now you're ready to try the shape table in a program. Some new commands you'll need are SCALE (how large to draw the man - I'll use 5, you experiment), ROTation (use 0 unless you want him to lean onto his side), DRAW (just what you'd expect) and XDRAW(erases).

```
10 HOME
20 HGR
30 HCOLOR=7
40SCALE=5
50 ROT=0
60VTAB22
70 PRINT "SHAPE TABLE DEMO"
80 PRINT CHR$(4);"BLOAD MARCHING MAN.PIC"
90 REM DRAW FIRST MAN IN CENTER OF SCREEN
100 DRAW 1 AT 140,80
110 FOR I=1TO500:NEXTI
120 REM THAT WAS A DELAY SO YOU CAN SEE
130 REM EACH STEP OF THE ANIMATION
140 XDRAW 1 AT 140,80
150 REM DRAW SECOND POSITION
160 DRAW 2 AT 140,80
170 FOR I=1TO500:NEXTI
180 XDRAW 2 AT 140,80
190 DRAW3 3 AT 140,80
200 FOR I=1 TO500: NEXTI
210 XDRAW 3 AT 140,80
220 DRAW 2 AT 140,80
230 FOR I=1TO500:NEXTI
240 XDRAW 2 AT 140,80
250 GOTO100
260 END
```

THIS IS AN INFINITE LOOP, SO THE MAN WILL MARCH FOREVER. FOR MORE INFORMATION, CONSULT THE APPLESOFT TUTORIAL, APPLE // USER'S GUIDE (LON POOLE), OR MICROCOMPUTER GRAPHICS (ROY MYERS).

LIBRARIAN'S REPORT

DOM sales in December totaled 60 and 335 library copieswere made! Thecombined general meeting and core meet at the Green Center was a big successagain this year as it has been in the past. Even though you may have copied lots of software, don't forget that we can now copy past DOMs over 12 months old for the standard $1.00 copy fee. We encourage you to complete your set of Apple Pi DOMs by library copying.

For those of you who may have missed the announcement at the December general meeting, Bob Garth has placed his ProTree BBS system in the Public Domain. It may be copied from the library, and instructions are included on the disk for purchasing documentation. This is the same software that was demonstrated a few months back by Bob at a general meeting. Copies of that system have also been sent to the International Apple Core and the Big Red Apple Club and I think you will soon find the ProTree BBS system all over the country. So thanks again to Bob Garth for enriching Apple Public Domain Software.

The January DOM is primarily a collection of utilities and includes the programs on the label below. The two major programs were donated by yet another Apple Pi-er, David Shaver. In fact, he donated an entire disk which is our new Utility Disk 13. His programs are Compactit, which will compact an entire DOS 3.3 disk into one or more binary files for easy modem transfer, and Sector Editor, which is a disk zap program for DOS 3.3 with very advanced features.

Also, the DOM contains a full-featured text editor and several utilities for working with text files. A section of catalog utilities provides five programs including another master catalog type program and a program to capture a disk directory into and ASCII text file which can then be edited with your text editor. A section of menu programs provides three additional menu programs besides the one used of late on DOMs.

For the graphics users, there is a pair of programs that explain and demonstrate how to use an included HIRES character generator. If you are learning to use the RWTS routine, there is an Applesoft implementation of a sector reader which should be very instructive. If you need an Integer BASIC loader, there is a program to BRUN before using an Integer program which temporarily loads a working version of Integer BASIC. Finally, there are a pair of programs which identify the cards in your machine and the type of machine you are using. And besides, these, there are twelve other programs.

The reverse side of the DOM contains the new ProDOS enhanced File Cabinet which is a public domain database program. It is an update of a classic Apple DOS 3.3 program that many of you have and I'm sure you will appreciate the ProDOS version. The DOS 3.3 version of Softgraph on the back of the November DOM has been corrected. If you want it replaced, the librarian will copy a working version for you.

See you at the general meeting!

-- Ed Thompson, Software Librarian --

```
Denver Apple Pi's Disk of the Month
#   66      JAN   1986    DOM
APPLE PI DOM CONNECT DOTS CATALOG CAPT CATALOG IMPR CATALOG    I
CATALOG UN-I THE CAT      TEXT EDITOR  TEXT EDITOR  TEXT FILE RA
TEXT FILE RE TLIST TEXT F MENU         MENU CLC     MENU EGNER
MENU OLD APP BASE CONVERT COMPACTIT V1 EPSON MAILIN FILEDUMP
FP ABORT     FP LISTER.EX FPLIST.INSTR FP LIST SET  FREE SECTORS
HIRES CHARS  HIRES CHARAC INTEGER BASI MACHINE IDEN PRGSTAT.INST
PROGSTAT     READ SECTOR  RESET DISABL RESTORE DELE SECTOR EDITO
SPEED UP DOS SYSTEM CONFI ARENA        CAT1         CAT2
RUNPACK      SHAPE1
```

SEED NEEDS EDITOR

The Apple Pi Seed has gone without an editor for several months now. A newsletter doesn't just happen by magic, and the few people who have been getting the Seed to you each month are very overburdened and cannot continue to put in all the hours demanded by their regular offices and do the Seed as well. An editor is needed NOW!

What do you need to be an editor? A desire to contribute some time to Denver Appe Pi each month. You do not need to be a printer or graphics person; that is well established. All you need is a flair for writing and good people skills. You need to gather articles from the monthly regulars like the librarian and president, and then find enough interesting stuff to fill the rest of the pages.

This is your big chance! Come on, let us know that you want to be Editor of the Seed!

JOHNNY APPLESEED

Dear John,

I have heard that therre is a company that can save a disk if you've spilled coffee on it. Is this true?
Clumsy

Dear Clumsy,

Yes, in most instances you can retrieve the data on a disk if you have spilled things on the disk. If the disk is physically damaged, or was near a magnet, then this is not the case. Polaroid (the camera people) have recently become a disk vendor and offer a data retrieval service to anyone who has a mishap while using a Polaroid disk.

John

Dear John,

Who is the biggest-selling software made for, Apple or IBM?

Just Wondering

Dear Just Wondering,

The biggest sellers are dependent on what type of software it is. For instance, business software tends to be for IBM, and personal software is mostly for Apple or Commodore. Here is a breakdown of the Softsel Hot List for the last week of December 1985:

Business top 5:

1.	1-2-3 -Lotus	IBM
2.	Turbo Lightning & Lib. -Borland	IBM
3.	Reflex -Borland	IBM
4.	Sidekick -Borland	IBM/MAC
5.	1-2-3 Report -Lotus	IBM

Recreational top 5:

1.	Gato Spectrum Holobyte	AP/IBM/MAC
2.	Wizardry -Sir-tech	AP/IBM/MAC
3.	Flight Simulator // -Sublogic	AP/COM/AT
4.	Microsoft Flight Simulator	IBM
5.	Jet -Sublogic	IBM/COM

Home software top 5:

1.	Print Shop -Broderbund	AP/IBM/COM/AT
2.	Newsroom -Springboard	AP/IBM/COM
3.	Print Shop Companion	AP/IBM/COM/AT
4.	Bank Street Writer	AP/IBM/COM/AT
5.	Print Shop Graphics Lib.3	AP/IBM/COM/AT

John

Dear John,

When I boot an old DOS 3.3 system master on my //c it prints that I am loading the language card. What is the language card?

Bilingual

Dear Bilingual,

In the old days of the Apple][, in order to get 64K you had to add a RAM memory board that had 16K of RAM, and it was in this area of memory that the language, Integer BASIC, was stored. When the //e and //c came out, this add-on was no longer needed, but the message still appears when booting an older system master.

John

ProDOS TECHNICAL NOTE #13

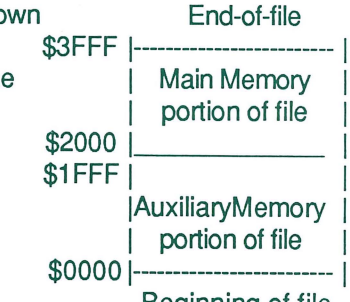
Double High Resolution Graphics Files

(6 January 1984)

The 128K Apple //e supports a graphics mode known as Double Hi-Res Graphics in which both main and auxiliary memory hi-res graphics pages are used to produce pictures with twice as many dot positions horizontally.

Apple /// graphics has a similar mode and a FOTOFILE file type ($08) has been defined under SOS to contain the screen image. All 16K double hi-res files under ProDOS should be of this file type.

The format of the file is as shown at the right.The "Graphics stored in the 121st byte of the file(Location $78 in the file). The modes for both 1st and 2nd page ofdouble hi-res are:



	Pg 1	Pg2
280 X 192 Limited Color =	1	5
560 X 192 Black and White =	2	6
140 X 192 Full Color	3	7

The normal Apple][hi-res 280 X 192 screen may be BSAVED as usual. If you desire, for Apple /// SOS compatibility, you may also save these screens as an 8K type $08 FOTOFILE and mark the graphics mode as zero (page 1) or four (page 2), (Apple /// 280 X 192 Black and White mode).

MESSAGE CENTER!

273-1150

ProDOS TECHNICAL NOTE #14

Selector/Dispatcher Conventions

(revised 09 March 1984)

ProDOS MLI call $65, the QUIT call moves addresses $D100 - $D3FF from the second 4K-byte bank of RAM of the language card to $1000 and executes a JMP to $1000. What initially resides in that area is OUR dispatcher code.

The dispatcher once executed does the following:

1. Interactively allows you to enter a prefix and file name of the system program (interpreter) that you wish to execute.

2. Stores the system program name at $280 starting with a length byte. This is done so once the system program executes, it can find from where it was started and locate any files it could need for processing.

3. Closes any open files.

4. Clears the bit map and protects the zero, stack, text and ProDOS Global pages.

5. Reads in the system file at $2000 and executes a JMP to $2000.

If you wish, you can install your own QUIT code which may load in your own full blown selector program. If you choose to do this, you must at some point:

1. Follow steps 2 - 4 above.

2. THE $D100 BYTE MUST BE A CLD ($D8) INSTRUCTION. This convention is established so programs will be able to tell whether it is selector code or the ProDOS dispatcher code that is resident.

In addition to just leaving the pathname at $280 for the interpreter's own use, a method to enable a selector program to specify an accompanying 'STARTUP' program has been defined. Once active, an interpreter can immediately run that program.

The procedure will be to reserve an area in the system file which will be overwritten by a selector program with the 'STARTUP' program's name. The interpreter would then load and execute that specified program.

The actual nuts and bolts of this procedure are as follows:

The selector program will look at the first byte of the interpreter at $2000. If it is a JMP ($4C) instruction, and bytes $2003 and $2004 are both $EE's, then byte $2005 will be interpreted as a buffer size indicator with the buffer starting at $2006. The string at $2006 would be the normal ProDOS pathname or partial pathname starting with a length byte.

JMP CONT	$2000 - $2002
$EE | $EE	$2003 - $2004
$41 (eg.)	$2005
$07 | $2006	
STARTUP	$2007 - $200D
:	
CONT: (eg.)	$2047

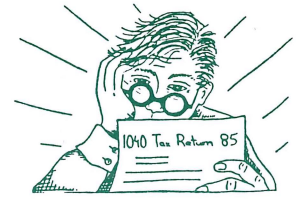
The two $EE's serve as a marker to the selector program to let it know that this particular interpreter can run a startup program. The interpreters that will support this feature will of course supply their own default string which may be a startup program or a flag of your own choice.

For more information on Interpreter Conventions, please see ProDOS Technical Note #7.

This is the last Tech. Note from Apple on ProDOS.

Other Tech. Notes will be in the Seed.

Tired Of The Tax Hassle?
Peopleware Has The Answer
TAXPAK-86



TAXPAK is an interactive income tax preparation program designed and priced for home use. It will guide you through the preparation of your 1985 (due in 1986) Form 1040 and Schedules A, B, C, D, G, SE, and W. It will calculate all subtotals, and perform all tax calculations automatically. When you are done, it will print the return in a format that you can send to the IRS.

TAXPAK is easy to use. It gives you an on-screen facsimile of the tax form, and provides full-screen editing capabilities, so that you can easily enter or change any amounts.

TAXPAK is fun. It operates like a tax spreadsheet, only it is much faster. Whenever you make an entry, all affected lines of the return are recalculated and updated within one second. Information is automatically transferred from form to form. A what-if feature allows you to explore different tax alternatives.

TAXPAK is informative. A unique feature of TAXPAK is that the bottom line of the display screen is used as a status line. It will tell you how much tax you owe, or how big your refund will be. It will also tell you how much your tax liability was changed by the last entry you made. It shows your total income, adjusted gross income and taxable income. And it gives your total tax amount and your tax bracket. All this information is updated and redisplayed after each entry you make. You see the effect of every line of the return on your total tax picture, as you enter it.

System requirements: TAXPAK is available for the IBM PC, PC/XT, PCjr and most compatibles, 128k, and DOS 2.0. Apple II version runs on any Apple II with 64k and DOS 3.3 CP/M version available in Kaypro, Osborne and most other formats (specify when ordering), 64k and DOS 2.2. Color monitors and hard disk drives are supported.

I am interested in TAXPAK-86 for: ☐ Apple II ☐ IBM PC ☐ CP/M

Send me: ☐ TAXPAK-86 for $39.95 ☐ TAXPAK Trial Disk for $3.00 Format _____

Name: _____ ☐ Visa ☐ MC ☐ Check

Address: _____ Accnt # _____

City _____ State: _____ Zip: _____ Expiration Date _____

Add $1.00 shipping to all orders. Colorado residents add 4.1% sales tax. Phone Orders:

Mail to: Peopleware, 10762 Twin Spruce Gap, Golden, CO 80403 (303) 642-0101

AUCTION

Apple Pi is having an auction at the March general meeting (March 6th)> We will have new and used items from our store members and items from the general membership. If you have any items, hardware or software, that you want to sell at the auction, please pick up a registration form at any Apple Pi meeting. A small percentage of the purchase price will be charged to pay for the auctioneer and club, and the rest of the money will go to the seller of the goods. If you are a buyer, don't miss this meeting!

HELP WANTED:

APPLE PI IS NUMBER 1

IT'S PEOPLE LIKE YOU HELPING PEOPLE LIKE YOU!

THE SEED

denver apple pi

P.O. Box 17467
Denver, CO 80217

BULK RATE
U.S. POSTAGE
PAID
DENVER, CO
PERMIT- 3182